Introduction to NGINX

A Professional Web Server for Real Applications

- So far, we've used the built-in PHP development server (php -S) to host our PHP web applications.
- While convenient for learning and testing, it has many limitations and is not suitable for production.
- To build real-world web applications, we need to use a proper web server.
    - In this section, we'll explore NGINX, a high-performance web server commonly used with PHP.
    - Another widely used alternative is Apache, which is also popular in many PHP environments.

What is NGINX?

NGINX (pronounced "Engine-X") is a powerful, high-performance web server used by:

- 🌐 **Netflix** - Video streaming
- 🛒 **Airbnb** - Travel platform
- 📚 **GitHub** - Code hosting
- 💬 **Slack** - Team communication
- 📊 **Dropbox** - File storage

Key Features

- ⚡ **High Performance**: Handles thousands of connections
- 🔀 **Reverse Proxy**: Routes requests to backend applications
- 📁 **Static Files**: Serves images, CSS, JS efficiently
- 🔧 **Load Balancing**: Distributes traffic across servers
- 🛡️ **Production Ready**: Used in real-world applications

NGINX vs PHP Built-in Server

Feature	PHP -S	NGINX
Purpose	Development only	Production ready
Performance	Limited	High performance
Concurrent Users	Few	Thousands
Static Files	Basic	Optimized
Configuration	None	Highly configurable
SSL/HTTPS	Not supported	Full support
Real World Usage	❌ Never	✅ Everywhere

Goal: Install NGINX and learn professional web server setup!

Installation NGINX

1. **Windows** - Using Chocolatey (recommended) or manual download
2. **macOS** - Using Homebrew (easiest method)
3. **Linux** - Using package managers (apt/yum)
4. **Verification** - Confirming installation works

Prerequisites

- Administrator/sudo access
- Basic command line knowledge
- No existing web servers on port 80 (stop Apache if running)

Windows Installation

Method 1: Using Chocolatey (Recommended)

Step 1: Install Chocolatey

Open **PowerShell as Administrator** and run:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072;
iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

Step 2: Install NGINX

```
choco install nginx
```

Step 3: Start NGINX

```
# Navigate to NGINX directory (usually C:\tools\nginx)
cd C:\tools\nginx
.\nginx.exe
```

Method 2: Manual Download

Step 1: Download NGINX

1. Go to http://nginx.org/en/download.html
2. Download **nginx/Windows** (stable version)
3. Extract to `C:\nginx` (create folder if needed)

Step 2: Start NGINX

Open **Command Prompt as Administrator**:

```
cd C:\nginx
nginx.exe
```

Step 3: Verify Installation

Open browser and go to: **http://localhost**

You should see: "Welcome to nginx!"

macOS Installation

Using Homebrew (Easiest Method)

Step 1: Install Homebrew (if not installed)

Open **Terminal** and run:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Step 2: Install NGINX

```
brew install nginx
```

Step 3: Start NGINX

```
# Start NGINX
brew services start nginx

# Or start manually
nginx
```

Step 4: Verify Installation

Open browser and go to: **http://localhost:8080**

You should see: "Welcome to nginx!"

Linux Installation

Ubuntu/Debian (using apt)

Step 1: Update Package List

```
sudo apt update
```

Step 2: Install NGINX

```
sudo apt install nginx
```

Step 3: Start and Enable NGINX

```
# Start NGINX service
sudo systemctl start nginx

# Enable auto-start on boot
sudo systemctl enable nginx

# Check status
sudo systemctl status nginx
```

Verification Steps

1. Check if NGINX is Running

Windows:

```
tasklist | findstr nginx
```

macOS/Linux:

```
ps aux | grep nginx
```

2. Test Web Server

Open browser and navigate to:

- **Windows/Linux**: http://localhost
- **macOS**: http://localhost:8080

3. Check NGINX Version

```
nginx -v
```

You should see output like: `nginx version: nginx/1.18.0`

Basic NGINX Commands

Starting and Stopping NGINX

Windows:

```
# Start
nginx.exe

# Stop
nginx.exe -s stop

# Reload configuration
nginx.exe -s reload
```

macOS:

```
# Start
brew services start nginx
# or: nginx

# Stop
brew services stop nginx
# or: nginx -s stop

# Reload
nginx -s reload
```

Linux:

```
# Start
sudo systemctl start nginx

# Stop
sudo systemctl stop nginx

# Restart
sudo systemctl restart nginx

# Reload configuration
sudo systemctl reload nginx
# or: sudo nginx -s reload

# Check status
sudo systemctl status nginx
```

Important File Locations

Configuration Files

Windows (Chocolatey):

- Main config: `C:\tools\nginx\conf\nginx.conf`
- HTML files: `C:\tools\nginx\html\`

Windows (Manual):

- Main config: `C:\nginx\conf\nginx.conf`
- HTML files: `C:\nginx\html\`

macOS (Homebrew):

- Main config: `/usr/local/etc/nginx/nginx.conf`
- HTML files: `/usr/local/var/www/`

For Apple Silicon Mac:

- Main config: `/opt/homebrew/etc/nginx/nginx.conf`
- HTML files: `/opt/homebrew/var/www`

Linux:

- Main config: `/etc/nginx/nginx.conf`
- HTML files: `/var/www/html/`
- Site configs: `/etc/nginx/sites-available/`

Testing Your Installation

1. Create a Test HTML File

Navigate to your NGINX HTML directory and find the `index.html`.

You can also create `test.html`:

```
<!DOCTYPE html>
<html>
<head>
    <title>NGINX Test</title>
</head>
<body>
    <h1>NGINX is Working!</h1>
    <p>Congratulations! You've successfully installed NGINX.</p>
    <p>Date: <span id="date"></span></p>
    <script>
        document.getElementById('date').textContent = new Date().toLocaleString();
    </script>
</body>
</html>
```

2. Test the File

Visit: **http://localhost** (or **:8080** on macOS)

Visit: **http://localhost/test.html** (or **:8080** on macOS)

Troubleshooting Common Issues

1. Port Already in Use

Error: `bind() to 0.0.0.0:80 failed`

Solution: Another service is using port 80

```
# Check what's using port 80
netstat -tulpn | grep :80

# Stop Apache if running
sudo systemctl stop apache2  # Ubuntu
sudo systemctl stop httpd    # CentOS
```

2. Permission Denied (Linux)

Error: Permission denied

Solution: Use sudo or check the firewall

```
sudo nginx
sudo systemctl start nginx

# Check firewall
sudo ufw allow 'Nginx Full'  # Ubuntu
```

3. NGINX Won't Start (Windows)

Solutions:

- Run Command Prompt as Administrator
- Check if port 80 is free
- Disable Windows World Wide Web Publishing Service:

```
net stop w3svc
```

4. Configuration Errors

Test configuration before starting:

```
nginx -t
```

Common fix: Check syntax in `nginx.conf`

5. Can't Access from Other Devices

Solution: Configure firewall to allow HTTP (port 80)

Success Checklist

✅ **NGINX installed** using the appropriate method for your OS
✅ **NGINX running** - can see welcome page
✅ **Basic commands work** - start/stop/reload
✅ **Test file created** - custom HTML page loads
✅ **Version check** - `nginx -v` shows version number

If all are checked, you're ready for the next step.

Configuring NGINX to work with PHP!

Quick Start Summary

Installation Commands

```
# Windows (Chocolatey)
choco install nginx

# macOS
brew install nginx && brew services start nginx

# Ubuntu/Debian
sudo apt update && sudo apt install nginx && sudo systemctl start nginx

# CentOS/RHEL
sudo yum install nginx && sudo systemctl start nginx
```

Key Takeaways

Why NGINX Matters

1. **Industry Standard**: Used by major websites worldwide
2. **Performance**: Handles many more users than the PHP built-in server
3. **Production Ready**: What you'll use in real applications
4. **Professional Skills**: Essential for web developers

What You've Learned

- How to install NGINX on any operating system
- Basic NGINX commands and file locations
- How to verify your installation
- Common troubleshooting steps

Coming Next: Setting up NGINX with PHP for a complete web server stack!